LA-UR-04-7315 December 17, 2004

Rule-based Modeling of Biochemical Networks

JAMES R. FAEDER*, MICHAEL L. BLINOV*, BYRON GOLDSTEIN,

AND WILLIAM S. HLAVACEK

Theoretical Biology and Biophysics Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

Correspondence to: William S. Hlavacek, Mail Stop K710, T-10, Los Alamos, NM 87545. Tel: 505-665-1355 Fax: 505-665-3493 E-mail: wish@lanl.gov (The manuscript includes 58 pages total, 7 figures, and 2 tables.)

We present a method for generating a biochemical reaction network from a description of the interactions of components of biomolecules. The interactions are specified in the form of reaction rules, each of which defines a class of reaction associated with a type of interaction. Reactants within a class have shared properties, which are specified in the rule defining the class. A rule also provides a rate law, which governs each reaction in a class, and a template for transforming reactants into products. A set of reaction rules can be applied to a seed set of chemical species and, subsequently, any new species that are found as products of reactions to generate a list of reactions and a list of the chemical species that participate in these reactions, i.e., a reaction network, which can be translated into a mathematical model.

Key Words: local rules; automatic model generation; networks; signal transduction; combinatorial complexity; systems biology

*These authors contributed equally to the work.

INTRODUCTION

The cell is a complex adaptive system whose emergent behavior we understand only poorly. One reason for our lack of understanding is the complexity of cellular decision making, which is often mediated by a system of interacting proteins. Systems of interacting proteins are particularly prominent in signal transduction[1] [1], the focus of our interest here. Such a system is complex in part because the interactions of its constituent proteins generally have the potential to generate a large number of distinct chemical species [2, 3], which can be far greater than the number of proteins or protein interactions in the system. Moreover, these chemical species are generally interconnected through a network of even more numerous reactions. Two major sources of this complexity, which has been called combinatorial complexity for reasons that will become apparent, are catalytic interactions of proteins that lead to post-translational covalent modifications, such as enzyme-catalyzed phosphorylation of an amino acid residue in a protein substrate, and non-covalent associative interactions of proteins that lead to the formation of heterogeneous molecular complexes. Both types of interactions are common in the regulatory systems of a cell [1, 4, 5].

The magnitude of combinatorial complexity can be grasped through the following considerations. A protein that can be covalently modified at p sites (e.g., through phosphorylation of particular amino acid residues within particular protein motifs) can occupy up to 2^p modification states, and a protein that interacts with q binding partners (e.g., through the activities of protein interaction domains [5]) can occupy up to 2^q bound states. Binding- or modification-induced transitions between different conformational states add a further layer of complexity at the level of individual molecules. All of this complexity is multiplied for complexes. For an assembly of n proteins, the number of distinct possible configurations is on the order of $\Pi_{i=1}^{n} s_i$, where s_i is the number of possible states of protein i in the assembly. Thus, the number of chemical species in a system depends exponentially on the number of interactions in the system and may be quite large even when few interactions are involved. For example, a model of early signaling events mediated

[1]The physiological function of a signal-transduction network is to convert an environmental signal, such as the presence of an extracellular ligand of a cell-surface receptor, into cellular responses, such as changes in gene expression, which allow a cell to adapt to the demands of its environment.

by the immune recognition receptor FcϵRI includes 354 chemical species and 3680 unidirectional reactions, but these species and reactions arise from consideration of the interactions among only three signaling proteins—the multimeric receptor, FcϵRI, and two protein tyrosine kinases (PTKs), Lyn and Syk—and a ligand [6, 7].

The problem of combinatorial complexity has been largely ignored, but it is difficult to avoid if one wishes to model a signal-transduction network at the level of protein interactions, which is the level of detail accessed in experiments (e.g., one may introduce a mutation that blocks a particular protein-protein interaction) and the level of detail desired for mechanistic modeling [8]. Interactions between proteins are mediated by submolecular components, such as protein motifs or protein interaction domains, and most signaling proteins contain more than one such component. Thus, to track protein-protein interactions, one generally needs to account for multivalent interactions (for an example of one approach to this problem, see [9]). Further complexity arises from the dependence of protein interactions on molecular context, which often influences the enzymatic and binding activities of proteins. For example, the rate of an enzymatic reaction may depend on the co-localization of an enzyme and its substrate, and the binding of two interacting proteins may depend on the phosphorylation state of one or both of these proteins. Thus, in the absence of information about which species are important and unimportant, a mechanistic model of a signal-transduction network would ideally account for all the possible states of molecules that may have multiple states and all the possible multi-component complexes of molecules in a system.

Modeling a system marked by combinatorial complexity is problematic simply because of the large numbers of chemical species and reactions one may wish to include in a model. Models cannot be written by hand. Manually writing a mass-balance equation for each of the chemical species in a large reaction network would be far too time consuming and error prone. This barrier to modeling signal transduction, and other biological networks, has been recognized by a number of researchers (for example, see [3, 8, 10, 11, 12, 13]), and there have been some attempts to overcome it. One example is StochSim [14, 15, 16], a software tool for agent-based modeling of signal transduction. This tool addresses the problem of combinatorial complexity by treating

molecules as interacting software objects. Molecules that may have multiple states are handled; however, the ability to handle complexes is limited.

To deal with the problem of combinatorial complexity, we have developed a modeling approach that relies on the specification of reaction rules, which serve as generators of chemical reactions. The rules that comprise the specification of a model are associated with the possible interactions and transformations of the domains of molecules in a system. An example of such a domain is the Src homology 2 (SH2) domain, which is one of many conserved modular polypeptide chains that mediate protein-protein interactions [5]. A rule has essentially the same form as a chemical reaction (e.g., $A + B \stackrel{k}{\rightarrow} C$), but the rule provides a template that defines many different individual reactions, which form a reaction class. Within a reaction class, reactants undergo a common type of transformation and all reactants share certain properties, which are specified in the governing reaction rule. Only chemical species with these properties qualify as reactants. Any properties that are unspecified in the rule are assumed not to affect the mechanism or rate of a reaction, which is a simplification, but one that can be tested and refined as necessary for consistency with experimental observations. A set of reaction rules can be evaluated automatically (as we will describe here in detail) to derive a reaction network, which can then be converted into a mathematical model, such as a system of coupled ordinary differential equations (ODEs). However, the reaction network is the crux of the matter; once a network is available, it can be used as the basis for many types of mathematical models. The network is comprehensive for the scope of interactions considered, which is precisely defined by the reaction rules used to generate the network. The validity and usefulness of this approach relies largely on the modularity of protein domains. One must be able to associate interactions and transformations of these domains with classes of reactions in which only certain properties of reactants influence the rate of reaction.

The first implementation of our rule-based approach to modeling was *ad hoc*, being useful only for generating a particular model, the model mentioned above for FcϵRI signaling [6, 7]. Later, we generalized the implementation and developed general-purpose software called BioNetGen [17]. This software can be used to generate models for a variety of signal-transduction networks. How-

ever, the software in original form has a number of limitations. Reaction rules can be evaluated only after all possible chemical species are enumerated, which is a limitation when it is either undesirable or impossible to consider all these species (e.g., as when the number of possible species exceeds that number of molecules available to populate the species). Also, a reaction network must be generated in its entirety before it can be used to predict the dynamics of the network. Another limitation concerns the number of multi-state molecules in a complex: only two multi-state molecules may combine to form a complex within the limitations of the original representational and rule-processing capabilities of the software. Here, we present algorithmic improvements that remove these limitations. An updated version of BioNetGen (version 1.1) is available at our web site [18]. We also suggest here extensions of the methodology that will be needed if graphs are used to specify models as recently proposed [19]. The proposed graph-based conventions for model specification, which were inspired by the use of graphs to model chemical systems [20, 21], are intended to allow the connectivity of multi-component complexes to be represented systematically and explicitly.

1 OVERVIEW OF METHOD

The method for generating a model of any system, as presented here, consists of two distinct procedures: 1) specification of the model and 2) interpretation of this specification. Figure 1 illustrates specification of a system consisting of a ligand, a receptor, an adapter, and a kinase. Definitions of key terms that we will use in our discussion of rule-based modeling are provided in Fig. 2. The conventions of model specification provide a compact format for archiving and exchanging models [17, 19], i.e., an unambiguous language for encoding knowledge of a biological system, which is needed [22, 23]. The interpretation of a model specification depends on software that implements the procedures described later. The main result of interpreting a model specification is a chemical reaction network, which can be translated into a mathematical or computational model.

In broad outline, a model is specified as follows. One first identifies the molecules, components

of molecules, and possible states of components to be considered. Then, one specifies a reaction rule for each type of reaction to be considered. These specifications are sufficient to define the structure of a reaction network, but additional steps would be necessary to create a predictive mathematical model. One might wish to specify the forms of rate laws in reaction rules (e.g., elementary or Michaelis-Menten rate laws), the values of kinetic parameters, and the numbers of molecules of each type in a system. To translate the specification of a model into a reaction network, one begins by specifying a seed set of chemical species. After this step, the reaction rules are applied automatically and iteratively, starting with the seed set of chemical species. In this procedure, chemical species that qualify as reactants are identified, and then reactions and their products are generated. The products may include new chemical species. The procedure continues until a termination condition is satisfied. The default condition is exhaustive generation of all possible species and reactions given a set of rules and a set of initial species. These steps generate a list of reactions and a list of the participating chemical species, which can be used to obtain different types of models that predict system behavior.

2 REPRESENTING A SYSTEM

We have proposed two sets of conventions for specifying rule-based models. The first is algebraic, abstract, and text-based [17], whereas the second is visual, intuitive, and graph-based [19]. Below, we will describe the text-based conventions and methodology in detail. One disadvantage of the text-based approach is that the connectivity of a complex is not represented and must be handled on a case-by-case basis by the user. Furthermore, not all complexes can be represented in this way, and there are thus some models that cannot be constructed. Graphical conventions for specifying a model [19] are more powerful, as they represent a generalization of the text-based conventions, but also more difficult to implement. Although we expect the graphical conventions to eventually supercede the text-based ones, the issues that arise in implementing both sets of conventions are similar, and therefore, an extended discussion of the techiques we have applied is useful at this

time.

2.1 The Molecular Parts, Their States and Complexes

Chemical species are represented using text strings. A simple name (e.g., A) may be used to represent any particular individual chemical species. A species represented this way is called a single-state species. A molecule string (Fig. 2), which comprises a name and an ordered list of indices, may be used to represent a particular species or set of related species containing a particular molecule. The molecule may have multiple components, and each component may have multiple states[2]. There is an index for each component, and this index indicates the state of the component or a range of possible states. A species represented by a molecule string is called a multi-state species. A list of molecule strings may be used to represent complexes containing particular molecules. The conventions of representation are elaborated below.

The indices of a molecule string form an ordered list, each element of which has a fixed position. If the indices of a molecule string are all integers, then the string represents a particular chemical species that contains a molecule, which is indicated by the name of the molecule string. Each index is associated with a component of this molecule. The possible states of each component are associated with integer values, which range, by convention, from 0, for the first state, to $n-1$, for the nth and final state. Molecular components and their states can be associated with descriptive names in comment lines of an input file, but these names are optional and they are not used in the process of network generation. All possible states of a component can be referenced by a wildcard character, *. If the wildcard is included among the indices, then a molecule string represents a set of chemical species. A list of period-separated molecule strings that have integer-valued indices, such as $R(1,1).R(1,1)$, specifies a particular multi-component complex containing a set

[2]Component states can be introduced to represent different conformations or modified forms of a molecular component. For example, the enzymatic activity of a kinase domain might be upregulated by phosphorylation of its activation loop, which causes a conformational change. To distinguish the inactive and active forms of such a kinase, we need to track its conformational state, or equivalently the phosphorylation state of its activation loop. An alternative to introducing a phosphorylation state would be to represent a phosphate group as a distinct component. If this approach is followed, it is important to distinguish between covalent and non-covalent bonds when specifying a model.

of molecules (two molecules of R in the example). If a molecule string in such a list includes a wildcard, which references multiple states, then the list represents a set of complexes. Note that the connectivity of components within a complex is not explicitly represented, which is a limitation of the current representational scheme.

BioNetGen requires all single- and multi-state species used in reaction rules to be declared in an input file. A single-state species declaration introduces the alphanumeric name (e.g., A) to be used for a particular single-state species. A multi-state species declaration introduces a set of molecule strings, all with the same alphanumeric name. The declaration consists of this name, followed by a list of integer numbers. The length of this list specifies the number of indices to be considered for each molecule string (i.e., the number of components of a molecule), and the value of each integer in the list specifies the number of values to be allowed for the index at the same position in the list of indices of each molecule string (i.e., the number of possible states for a component). An example of a multi-state species declaration is $R\ 2\ 4$, which introduces a molecule string R with two indices, i.e., a molecule with two components. The first component has two possible states, which are taken to be 0 and 1 by convention, and the second component has four possible states (0, 1, 2, and 3). Thus, the molecule strings (and corresponding multi-state species) that are introduced by the declaration $R\ 2\ 4$ are $R(0,0)$, $R(0,1), \ldots, R(1,3)$. If an undeclared multi-state species is generated during the process of network generation, an error message is reported.

A multi-component complex of molecules can be represented in one of two ways, both of which involve the use of at least one molecule string. In the first way, the representation of a complex is subsumed into the state description of a molecular component, which is typically a site of protein-protein interaction. For example, given two binding partners, A and B, we can use a multi-state species declaration, namely $A\ 2$, to introduce $A(0)$ and $A(1)$, which we can then take to represent the free form of A and the complex of A and B (or equivalently, the form of A with its first and only component in the bound state). The free form of B can be represented simply as B. The implicit representation of a physical complex using a molecule string, such as $A(1)$, requires that a user associate the complex with the state of a molecular component and specify

self-consistent reaction rules. The burden is on the user to devise an appropriate mapping between states and complexes, which can be cumbersome.

The second way a complex can be represented is as a period-separated list of molecule strings, such as $A(1,0).B(1)$ or $A(1,1).B(1).C(1,0,0)$. A list may contain any number of molecule strings. The order of these strings is unimportant for purposes of model specification. Thus, specifications of $A(1,0).B(1)$ and $B(1).A(1,0)$, for example, in an input file are equivalent. Declarations of complexes, like declarations of single- and multi-state species, may be included in an input file. These declarations, if used, define the complexes that are allowed in a system. The use of such declarations is discussed further in Sec. 3.4. An example of a complex declaration is $R(1,*).R(1,*)$, which introduces homodimeric complexes of R in which the first component of each molecule of R is in state 1 and the second component may be in any of its possible states, which are delimited by the multi-state species declaration of R. If this declaration is $R\ 2\ 2$, then the declaration $R(1,*).R(1,*)$ introduces three dimers: $R(1,0).R(1,0)$, $R(1,1).R(1,0)$, which is equivalent to $R(1,0).R(1,1)$ (see below), and $R(1,1).R(1.1)$.

In summary, there are three ways to refer to chemical species or sets of chemical species in a system. A user may use a simple name, a molecule string, or a list of molecule strings. Molecule strings may contain wildcards, which allow a user to refer to a set of species. A species is a member of a set, such as $R(1,*)$, if it is represented by a matching molecule string, such as $R(1,0)$ or $R(1,1)$. To define the species allowed in a system, a user may declare single-state species (e.g., B), multi-state species (e.g., $R\ 2\ 4$ and $A\ 2$), and complexes (lists of multi-state molecule strings), such as $R(1,*).R(1,*)$.

BioNetGen arranges molecule strings within a complex according to a predetermined sort order, which ensures that each complex is associated with a unique text string. Molecules within a complex are first sorted alphabetically by name (e.g., $A(1)$ before $B(1)$). Any molecule strings with the same name, such as $R(0,0)$ and $R(0,1)$, are then sorted according to the states of their components, which as discussed earlier, have a fixed order and are represented by integer-valued indices. For example, the first and second indices in $R(1,2)$ always correspond to the first and

second components of R. The states of components are compared from right to left. Thus, in a comparison of $R(0,0)$ and $R(0,1)$, we examine 0 and 1. Higher values take precedence over lower values. Thus, $R(0,0).R(0,1)$ is rewritten as $R(0,1).R(0,0)$. Another example of a canonical listing of molecule strings is $R(1,1).R(0,1).R(1,0).R(0,0)$. The canonical ordering of molecule strings facilitates comparisons of species and reactions. These comparisons are necessary to maintain unique lists of species and reactions in a network.

2.2 Reaction Rules for Molecular Interactions and Transformations

A reaction rule is defined for each class of reaction to be considered in a system. In cases we have considered, classes are associated with an interaction between two components, such as binding of a particular protein interaction domain to a binding site (e.g., the SH2 domain of one protein binding the phosphorylated tyrosine residue of a second protein). A reaction rule has the same form as a chemical reaction, e.g., $A + B \stackrel{k}{\rightarrow} C$ for an elementary bimolecular associative reaction with rate constant k. However, in a reaction rule, the reactants and products may be replaced by group patterns. A group pattern defines a set of chemical species that share a set of component states and bonds, which are specified in the group pattern. Group patterns in BioNetGen are regular expressions, i.e., string-matching patterns. More specifically, a group pattern is a simple name, a molecule string, or a list of such strings that is used to identify a single species or set of species. A group pattern often includes a wildcard.

To consider a simple example of a reaction rule in BioNetGen format, let us return to the example declarations of binding partners A and B discussed above (i.e., A 2 and B), but let us now consider molecule A to contain a second component, say a tyrosine residue, which has two states, unphosphorylated and phosphorylated. The appropriate multi-state species declaration for this scenario is A 2 2. If the phosphorylation state of component 2 in A does not affect the interaction of A and B, then we can write a reaction rule for binding of A and B as follows:

$$A(0,*) + B \stackrel{k_{AB}}{\longrightarrow} A(1,*) \tag{1}$$

where $A(0,*)$ and $A(1,*)$ are reactant and product group patterns and $*$ is a wildcard, which in this case indicates that component 2 of any species of type A can be in any of its possible allowed states. The reactant group pattern $A(0,*)$ identifies the chemical species that qualify as reactants (any species of type A with component 1 in state 0), and the product group pattern $A(1,*)$ indicates how reactants are transformed into products (the state of component 1 in a species of type A is changed from 0 to 1). The convention is that only transformations explicity indicated in a reaction rule occur. Thus, the state of component 2 in any species of type A is understood to be unaffected by reactions in the class of reactions defined by the rule in Equation (1), i.e., only the following reactions are generated:

$$A(0,0) + B \xrightarrow{k_{AB}} A(1,0),$$

$$A(0,1) + B \xrightarrow{k_{AB}} A(1,1).$$

In a BioNetGen input file, rules can be specified in uni- or bi-directional forms:

$$reactant\_1 + \ldots \rightarrow product\_1 + \ldots \; forward\_rate\_constant$$

or

$$reactant\_1 + \ldots \leftrightarrow product\_1 + \ldots \; forward\_rate\_constant, \; reverse\_rate\_constant$$

where $reactant\_1$, $product\_1$, etc. are group patterns (Fig. 2). Specifying a bidirectional reaction rule is equivalent to specifying two unidirectional reation rules. Any number of group patterns may be included in a rule. Group patterns can be in three distinct formats:

- The group pattern for a single-state species with name $molecule$ is just $molecule$.
- The group pattern for a multi-state molecule with name $molecule$ has the format $molecule(state\_1, \ldots, state\_N)$. Any $state\_i$ can be replaced with a wildcard, which permits a match to any state of the i-th component of $molecule$.
- The group pattern for a complex is a collection of group patterns of the type indicated above, which are joined using periods, e.g., $molecule\_1(\ldots).molecule\_2(\ldots)$.

Wildcards can also be used to select complexes of variable molecular composition:

- A group pattern ending with $*$ specifies a match to any complex that contains the preceding molecules. $A(*,1)*$, for example, would match both $A(0,1)$ and $A(1,1).A(1,0)$.
- A group pattern ending with $.*$ specifies a match to any complex that contains the preceding molecules and at least one additional molecule. $A(*,1).*$ would match $A(1,1).A(1,0)$ but not $A(0,1)$.

We refer to these non-state wildcards as molecular wildcards.

2.3 Rate Laws

Each reaction rule is associated with a rate law, which is taken to apply for all reactions within the class of reactions defined by the rule. In BioNetGen 1.1, the rate law is assumed to have the form of a rate law for an elementary reaction with the appropriate number of reactants, and as a result, only a rate constant is required in an input file to fully specify the rate law associated with a reaction rule. The rate law for a particular reaction in the reaction class defined by Eq. 1, the reaction wherein B binds to $A(0,0)$, would be

$$\text{rate} = k_{AB}[A(0,0)][B],$$

where square brackets are used to indicate concentrations. Higher order reactions may also be specified with

$$\text{rate} = \prod_{i=1}^{n} k[R_i],$$

where $[R_i]$ is the concentration of the i-th reactant. More complicated rate laws will be allowed in future versions of the software.

3 INTERPRETING REACTION RULES

Reaction rules are used, as described below, to identify chemical species that qualify as reactants and then to define reactions involving these reactants and the products of these reactions. This section describes in detail how reaction rules are parsed in BioNetGen and used to create new reactions and species from an existing set of species. The final subsection describes how the full reaction network is generated through the application of reaction rules to an initial set of chemical species. In the discussion that follows, we will assume group patterns in reaction rules involve only molecule strings. A simple name for a single-state species can be viewed as a special type of molecule string.

3.1 Establishing Correspondence Between Reactants and Products

The first step in processing a reaction rule is to establish correspondence between the molecule strings of reactants and products (Fig. 3). This correspondence is essential in defining the transformation of reactant species into product species. The default algorithm for assigning correspondence is as follows. Each molecule string in the group patterns on the reactant side is mapped (going from left to right) to the first molecule string specifying a molecule with the same name on the product side of the rule that is not already assigned a correspondence. Molecular wildcards are assigned a correspondence in the same way. A null correspondence is assigned if no match is found, which allows a molecule to be created or destroyed during a reaction. A null correspondence is not permitted for a molecular wildcard on the product side of a rule. Similarly, a wildcard for a component state in a product molecule is not permitted if the component state is specified in the corresponding reactant molecule. In other words, a component state cannot go from being defined on the reactant side to undefined on the product side.

Although this method of assigning correspondence is adequate for most reaction rules, certain types of rules require an alternative assignment of correspondence. Consider, for example, trying to define a reaction which involves the exchange of two molecules in two different complexes. One

might try to write a rule defining such a transformation to exchange two C molecules as follows

$$A(*).C(0,*) + A(*).C(1,*) \longrightarrow A(*).C(1,*) + A(*).C(0,*).$$

However, using default procedures, BioNetGen would assign correspondence in a way that would not generate the intended exchange. For example, given the reactants $A(0).C(0,0)$ and $A(1).C(1,1)$ it would generate the reaction

$$A(0).C(0,0) + A(1).C(1,1) \longrightarrow A(0).C(1,0) + A(1).C(0,1)$$

whereas the intended exchange would result in

$$A(0).C(0,0) + A(1).C(1,1) \longrightarrow A(0).C(1,1) + A(1).C(0,0).$$

BioNetGen 1.1 therefore provides a mechanism for assigning unique labels to molecules that allow a user to override the default assignment of correspondence. Labels are assigned on the reactant side of a rule by appending '%' to a molecule string followed by an alphanumeric label. These labels can then be appended to molecule strings on the product side to make explicit correspondences. The default algorithm is used to define corrspondences that are not explicitly declared in this way. Using this notation, the exchange reaction is represented by the rule

$$A(*)\%A1.C(0,*)\%C1 + A(*)\%A2.C(1,*)\%C2 \longrightarrow A(*)\%A1.C(1,*)\%C2 + A(*)\%A2.C(0,*)\%C1.$$

In this rule, the labels $A1$ and $A2$ are unnecessary, but demonstrate that each molecule in a complex can be assigned a label. Labels only apply to the rule in which they appear.

3.2 Generating Reactions

After correspondence is established for a reaction rule, the rule is applied to a list of chemical species. Each reactant group pattern is used to select a list of matching species, all of which are

possible reactants. A species may appear multiple times in a reactant list if there are multiple ways in which the molecule strings in the reactant group pattern can match the molecule strings of a species (examples are given below). Reactions are generated by looping over all possible sets of reactants drawn from the reactant list(s). Some sets are eliminated by filtering conditions (described in the next subsection) that take into account symmetry. For each matching set of reactants that passes through these filters, product species are generated from the reactant species using the correspondence between reactant and product molecule strings. For each product molecule string, the molecule matching the corresponding reactant molecule string is transformed into a product molecule by changing the component states to match those of the product molecule string. Component states that are specified by a wildcard in the product molecule string are unchanged. Once all product molecules have been generated, product complexes are generated (as needed) by concatenating the product strings as specified by product group patterns. Molecules within complexes are then rearranged into sort order (as described in Sec. 2.1), which guarantees a unique string representation for each chemical species.

After reactants and products are identified, a reaction string is generated, which consists of a comma-separated list of reactant species strings followed by a comma-separated list of product species strings and the corresponding rate constant. The reaction string is added to a temporary list of reactions generated by the application of this reaction rule to the list of chemical species. Once all possible reactant sets have been exhausted, this temporary list of reactions is added to a more global reaction list, as described in Sec. 3.4 below.

The possibility of reaction overlap, which occurs when two reactions with identical reactants and products are generated by different rules, is checked for and handled according to an optional user-defined precedence index. By default, all rules generate reactions with precedence index zero. When two reactions overlap, the reaction with the lower precedence index is deleted, but both reactions are kept if their precedence indices are equal. Precedence can be used, for examle, to define sub-classes of reactions with different rate constants or rate laws. This usage is illustrated in the BioNetGen input file `toy_coop.in`, which is available at our web site [18] and discussed

later (Sec. 7).

3.3 Reaction Symmetry and Multiplicity

Each reaction generated by a reaction rule is ultimately assigned an effective rate constant that, by convention, is the product of the rate constant specified in the rule and the multiplicity of the reaction, which is an integer that depends on reaction symmetry. The multiplicity of a reaction is 1 if there is only one reaction path from reactants to products, whereas the multiplicity is greater than 1 if there are multiple, indistinguishable reaction paths from reactants to products. Reactions for which the multiplicity is greater than 1 must be considered, for example, in models of multivalent ligand-receptor binding [24]. In the process of network generation, the multiplicity of a reaction has an initial value of 1 and is then incremented as indistinguishable instances of the reaction are generated, as described below. A user must be careful to specify rate constants in reaction rules that are consistent with the convention described above. Automatic determination of multiplicity is a critical feature of BioNetGen, because multiplicity can differ for reactions within the same reaction class.

Let us consider an example of a reaction with a multiplicity of two, which is generated by the following reaction rule for asymmetric dissociation of a dimer:

$$A(1,*).A(1,*) \xrightarrow{k_{-2}} A(1,*) + A(0,*).$$

When this rule is applied to the complex $A(1,0).A(1,0)$, either of two reactions can be generated, because either of the two molecules in the complex can be transformed into $A(0,0)$. The reactions can be written as

$$A(1,0).A(1,0) \xrightarrow{k_{-2}} A(1,0) + A(0,0),$$

$$A(1,0).A(1,0) \xrightarrow{k_{-2}} A(0,0) + A(1,0).$$

Of course, these reactions are chemically indistinguishable, which arises from the fact that the two molecules in the complex are identical. Thus, instead of two separate reactions, each parameterized by the rate constant k_{-2}, we can consider a single reaction, which is parameterized by an effective rate constant that accounts for the two indistinguishable paths to the reaction products:

$$A(1,0).A(1,0) \xrightarrow{k_{\text{eff}}} A(1,0) + A(0,0).$$

where $k_{\text{eff}} = 2k_{-2}$. In accordance with the convention, the effective rate constant is the product of the rate constant in the reaction rule, k_{-2}, and the multiplicity of the reaction, 2.

The procedure for determining the multiplicity of a reaction depends on the following definition. Two reactant molecule strings in a reaction rule are defined to be *equivalent* if and only if 1) they are identical and both appear in the same or identical reactant group patterns, and 2) their corresponding product molecule strings are identical and both appear in the same or identical product group patterns. The use of equivalence is explained below.

During the loop over reactant sets described above in Sec. 3.2, a set is discarded if either of the following two filtering conditions is applicable:

1. The order in which two equivalent molecule strings match molecules within separate species does not put the species in sort order (see Sec. 2.1).

2. The order in which two equivalent molecule strings match molecules within a single complex does not correspond to the order in which the molecules appear in the complex.

Both conditions prevent redundant reactions from being generated by imposing a particular order in which species and molecules must be matched by equivalent molecule strings.

An example of the application of Filtering Condition 1 is provided by processing the following reaction rule for a symmetric aggregation reaction

$$A(1,*) + A(1,*) \xrightarrow{k_{+2}} A(1,*).A(1,*).$$

The two reactant molecule strings are equivalent. Thus, when they match $A(1,1)$ and $A(1,0)$, respectively, BioNetGen generates the reaction

$$A(1,1) + A(1,0) \xrightarrow{k_{+2}} A(1,1).A(1,0).$$

However, when these molecule strings match $A(1,0)$ and $A(1,1)$, respectively, no reaction is generated because the first filtering condition applies.

An example of the application of Filtering Condition 2 is provided by processing the following reaction rule for a symmetric dimer breakup reaction

$$A(1,*).A(1,*) \xrightarrow{k_{-2}} A(1,*) + A(1,*).$$

Again, the two reactant molecule strings are equivalent. Thus, when they match $A(1,1)$ and $A(1,0)$, respectively, in the complex $A(1,1).A(1,0)$, BioNetGen generates the reaction

$$A(1,1).A(1,0) \xrightarrow{k_{-2}} A(1,1) + A(1,0)$$

However, when the molecule strings match $A(1,0)$ and $A(1,1)$, respectively, in the same complex, the second filtering condition applies and no reaction is generated. For comparison, consider the following rule for asymmetric dissociation of a dimer

$$A(1,*).A(1,*) \xrightarrow{k_{-2}} A(1,*) + A(0,*)$$

and let us apply this rule to the same dimer $A(1,1).A(1,0)$. In this case, the two reactant molecule strings are not equivalent because their corresponding product molecule strings are different. Thus, matching $A(1,1)$ and $A(1,0)$, respectively, generates

$$A(1,1).A(1,0) \xrightarrow{k_{-2}} A(1,1) + A(0,0),$$

and matching $A(1,0)$ and $A(1,1)$, respectively, generates

$$A(1,1).A(1,0) \xrightarrow{k_{-2}} A(1,0) + A(0,1).$$

As can be seen, both possible sets of products of $A(1,1).A(1,0)$ dissociation are properly generated.

3.4 Generating the Full Reaction Network

The full biochemical reaction network is generated through application of the reaction rules to an initial set of chemical species. This initial set may be defined explicitly by the user or generated automatically from declarations of single- and multi-state species and complexes.

A user-declared set of initial species is typically small, including, for example, only the unbound and unmodified forms of the molecules in a system, and is used as the starting point for iterative application of the reaction rules to generate new species and reactions. Each cycle of rule application involves applying every reaction rule to the set of species present at the beginning of the cycle (see below for a more detailed description). Rule applications can generate both new reactions and new species, which are added to the list of species in the network and used to generate new reactions in the next cycle. Iterative rule application is terminated if any of the following user-specified conditions is reached: a specified number of product species or reactions have been generated, a specified number of rule applications has been peformed (corresponding to a particular value of k in the notation below), or no new reactions are generated. By default, the last condition is used, which results in the generation of all species and reactions reachable from the initial set or, if polymer chain elongation reactions are possible [3], an infinite loop. An infinite loop can be avoided by overriding the default termination condition in the input file or, for example, by using declarations of complexes to limit their sizes. Iterative generation from a specified seed set of species does not require a complete specification of the species that are possible in a system, and only species accessible from the seed set are generated.

If the user does not declare a set of initial species, the following species are automatically generated and used as the initial set in network generation: all declared single-state species, all declared multi-state species with state indices that fall within the allowed ranges, and all declared complexes. In this mode of network generation, the initial set of species is considered to be the set of all possible species. Thus, any rules that generate complexes must be accompanied by an explicit declaration of the complexes (see Sec. 2.1). This prevents the accidental generation of unintended species, and provides a useful check that reaction rules are behaving as expected. Because the generation of additional species is forbidden, reaction rules need only be applied once to the initial set, which also helps to accelerate network generation.

The process of defining and generating a biochemical reaction network in BioNetGen is summarized by the following steps:

1. Identify the molecules, components of molecules, and possible states of components to be considered.

2. Specify a reaction rule for each type of reaction to be considered.

3. Provide an initial list of distinct chemical species $\mathcal{L}^0_{\rm species}$ (a seed set).

4. For each reaction rule, identify all sets of species in $\mathcal{L}^0_{\rm species}$ that qualify as reactants and define a reaction, as specified in the rule, for each unique set of reactants and products to obtain a list of distinct reactions $\mathcal{L}^0_{\rm reactions}$.

5. Identify chemical species that are products in the list $\mathcal{L}^0_{\rm reactions}$ but that are not in the list $\mathcal{L}^0_{\rm species}$ to obtain a list of new species $\mathcal{L}^1_{\rm species}$.

6. Starting with $k = 1$, apply each reaction rule to the set of all species $\bigcup_{i=0}^{k} \mathcal{L}^i_{\rm species}$ to generate all possible reactions in which at least one reactant is an element of the list $\mathcal{L}^k_{\rm species}$ to obtain, as described in Steps 4 and 5, a list of new reactions $\mathcal{L}^k_{\rm reactions}$ and a list of new product species $\mathcal{L}^{k+1}_{\rm species}$. Iterate, incrementing the integer k, until no new species are found or a specified termination condition is satisfied.

The specification of a model is completed after the first three steps. Interpretation of this specification is then performed in the subsequent steps, which end with a list of reactions implied by the model specification and a list of chemical species that participate in these reactions. Step 4 involves applying the reaction rules to generate all chemical reactions in which the seed set of chemical species can participate. This is the terminal point in network generation unless the user has explicitly declared a seed set of species. Iterative generation proceeds with Step 5, the identification of new chemical species generated by application of the reaction rules, followed by repeated cycles of new reaction generation (Step 4) and new species identification (Step 5).

4 FROM NETWORK TO MODEL

Once a reaction network has been generated, the next step is to translate the network into a mathematical model. Many different types of models can be obtained from a list of reactions and a list of the chemical species that participate in these reactions. Note that the different types of models may require additional information beyond the lists. For example, predictions of an ODE-based model require specification of the initial concentrations of chemical species and numerical values of rate constants.

One type of model that can be derived from a list of reactions is a graph. A graph representing a network can be analyzed to study static properties of the network, such as topological robustness [25]. It is straightforward to convert a list of reactions into a bipartite graph that represents a biochemical reaction network. In such a graph, one type of node corresponds to chemical species and the other type of node corresponds to reactions. Directed edges join nodes. An edge indicates that a chemical species participates in a unidirectional reaction. The direction of the edge distinguishes between reactant and product. Graphs have been used to represent and study protein interaction networks and other types of biochemical networks [25].

Another type of static model is a stoichiometric model, the essence of which is the stoichiometric matrix $\mathbf{S}$ [26]. The stoichiometric matrix is constructed from a list of reactions and participating

chemical species as follows. A row is added for every chemical species, and a column is added for every unidirectional reaction. Thus, the matrix has m rows and n columns for a network with m species and n unidirectional reactions. Each matrix element s_{ij} in row i and column j indicates the stoichiometry of species i in reaction j. This element is negative if species i is consumed in reaction j and positive if species i is produced in reaction j. Recently, stoichiometric analysis has been used in the theoretical study of signal transduction [27, 28].

It is straightforward to go beyond static models and derive a mathematical model for the dynamics of a reaction network. For example, for a network with m chemical species and n unidirectional reactions, we can write

$$\frac{d}{dt}\mathbf{x} = \mathbf{S} \cdot \mathbf{v}$$

where t is time, $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ is a vector of concentrations of the chemical species, $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T$ is a vector of fluxes through the reactions, each of which is given by a rate law in a reaction rule (e.g., kx_ix_j for a biomolecular elementary reaction with rate constant k and reactants i and j having concentrations x_i and x_j respectively), and $\mathbf{S}$ is the stoichiometric matrix. Once such a model is available, one can estimate parameters in rate laws and then carry out a computational analysis of the dynamic properties of the model. A BioNetGen input file may include a list of rate constants and their values and initial non-zero concentrations of specified chemical species.

5 SIMULTANEOUS NETWORK GENERATION AND SIMULATION

A new feature of BioNetGen 1.1 is the ability to implement a stochastic simulation algorithm (i.e., a Monte Carlo method for simulating discrete-event reaction kinetics) [29, 30, 31] at the same time a reaction network is being generated, as in [32]. As discussed elsewhere [3, 17], this feature (on-the-fly generation and simulation of a reaction network) may be useful when the number of

possible chemical species is exceedingly large, because only species that are populated enter into the calculations. Thus, many species and reactions may never need to be generated when network generation and simulation are coupled.

The method of on-the-fly network generation and simulation is accomplished by the following modifications of the six-step procedure described above. In Step 3, the seed set of chemical species $\mathcal{L}^0_{\rm species}$ must contain at least one species with non-zero population, and every species in the seed set must be assigned an initial value. Furthermore, the seed set of chemical species must include all of the chemical species that are populated at time $t = 0$ of the simulation. Steps 4 and 5 are then carried out, and any new species generated at the end of these steps are each assigned zero population number. At this point, $k = 1$. The process then continues based on a user-specified upper bound on the index k, which we will denote as k'. Any value can be assigned to k' so long as all chemical species in $\mathcal{L}^{k'}_{\rm species}$ have zero population. After Steps 4 and 5, any $k' \geq 1$ is a valid assignment. Given some k', the lists $\mathcal{L}^i_{\rm species}$ and $\mathcal{L}^{i-1}_{\rm reactions}$ are generated for all $i \leq k'$ (Step 6) and then species numbers are updated using the stochastic simulation algorithm (in principle, any other method of updating species numbers or concentrations may also be used) until a member species of $\mathcal{L}^{k'}_{\rm species}$ becomes populated for the first time. When this happens, Step 6 is repeated once or any specified number of times, and the value of k' is replaced with the value of k at the end of this procedure. Updates of species numbers are again performed as before until a member species of $\mathcal{L}^{k'}_{\rm species}$ becomes populated for the first time etc. The sets $\bigcup_{i=0}^{k'} \mathcal{L}^i_{\rm species}$ and $\bigcup_{i=0}^{k'-1} \mathcal{L}^i_{\rm reactions}$ are sufficient to implement the method of stochastic simulation (cf. [32]).

It should be noted that the algorithm described above is lacking with respect to computational efficiency for two reasons. First, the method may generate numerous species and reactions that are unnecessary, because all species and reactions in $\mathcal{L}^{k'+1}_{\rm species}$ and $\mathcal{L}^{k'}_{\rm reactions}$ are generated, at least, if just one species in $\mathcal{L}^{k'}_{\rm species}$ becomes populated. An improvement would be to generate new species and reactions more selectively [33], but the present method addresses the problem of a non-terminating polymer chain elongation reaction, which for practical purposes could easily prevent exhaustive enumeration of the full potentiality of a network, which is limited only by the number

of molecules in a system [3]. It also provides a means to generate only part of a network and determine if this portion of the network is sufficiently large to contain all populated species. This capability may be desirable when the potentiality of a network is enormous and most species in it are never populated. Second, the method of stochastic simulation implemented in the software is the direct method of Gillespie. More efficient algorithms are available [34, 35, 36, 37, 38, 39].

6 READOUTS

The output of a model simulation includes the concentrations of all species in a chemical reaction network. However, proper organization of this information may be needed to relate model variables to experimental readouts. For this reason, we have introduced output function evaluation rules. Such a rule includes a specified mathematical function of the properties of the chemical species that belong to a specified group, such as the sum of concentrations of all species in a group. Thus, the rule consists of a mathematical function, one or more group patterns, and a mapping of the properties of chemical species into variables of the function. At present, output functions of two types can be specified in a BioNetGen input file: one is a sum of concentrations and the other is a weighted sum of concentrations. Each output function evaluation rule has the following format:

$$output\_type\ group\_name\ group\_pattern\_1 \ldots group\_pattern\_\mathrm{N}$$

The $output\_type$ element must be either $species$ or $molecules$. The $group\_name$ element is a user-defined label for referencing the output function. The rest of the function evaluation rule consists of a list of one or more group patterns that identify a group of chemical species. The $output\_type$ element indicates the type of sum to be calculated. A $species$ sum adds up the concentration of each species that matches a group pattern in the output rule. A $molecules$ sum adds up the concentration of each species that matches a group pattern in the output rule weighted by the number of matches found. Note that group patterns ending with a wildcard or containing several multi-state molecules of the same type can generate multiple matches for the same complex. Note

also that readouts can be defined to provide sanity checks (i.e., tests for proper model specification). For example, output functions can be used to confirm that mass is conserved in a model.

7 EXAMPLES

The best-documented example of rule-based modeling is perhaps provided by the FcϵRI model [6, 7]. Another rule-based model that we have developed, and begun to analyze, is an extended form of the model of Kholodenko et al. [40] for early events in signaling by the epidermal growth factor (EGF) receptor (EGFR). The extended model incorporates the same scope of protein interactions as the original model but accounts for a broader range of the chemical species (as discussed elsewhere [3]) that are implied as being possible by these interactions. BioNetGen input files, consistent with the standards of version 1.1, are available at our web site [18] for these models. Some of the properties of these models are summarized in Table 1. Models have been developed using software tools related to BioNetGen, such as STOCHSIM [14, 15, 16] and Cellerator [41], and a model recently developed by Chakraborty and co-workers [42] provides another example of rule-based modeling. In the remainder of this section, we will discuss a rule-based model for the toy system of Fig. 1. This model is simpler than the models mentioned above, but captures many of their essential features.

A BioNetGen input file (`toy.in` [18]) that specifies a model for the toy system of Fig. 1 is illustrated in Figs. 4–6. The lines in the input file delimiting the range of species included in the model are illustrated in Fig. 4, lines that specify reaction rules are illustrated in Fig. 5, and lines that define and request different readouts of the model are illustrated in Fig. 6. Because the representation scheme used by BioNetGen is flexible, models consistent with Fig. 1 could be specified in other ways. Calculations based on the toy model and the parameter values given in Table 2 are shown in Fig. 7. Note that in this model, the kinase in the system is taken to be activated (i.e., its enzymatic activity is upregulated) by phosphorylation.

The model of the toy system illustrated in Figs. 4–6 predicts that very little kinase activation is

induced by ligand addition (see panels (a) and (b) in Fig. 7). Because the toy system is comprised of signaling proteins and interactions of the types found in signaling cascades involving Toll-like receptors (TLRs) [43][3], which play an important role in immunity [44, 45], we decided to explore how phosphorylation of the kinase in the system might be more strongly induced by ligand. We found that a model incorporating cooperative interactions between kinases in the same complex predicts much greater ligand-induced kinase phosphorylation (see panel (c) in Fig. 7). This form of the model is encoded in the BioNetGen input file `toy_coop.in` and is available at our web site [18].

Cooperativity is modeled by decreasing the rate constants for adapter-receptor dissociation, kinase-adapter dissociation, and receptor dimer breakup by 100-fold when two kinase molecules are present in a complex. This decrease of rate constants for particular reactions is imagined to arise from interactions between kinase molecules when these molecules are in the same complex. Cooperativity is introduced in the BioNetGen input file `toy_coop.in` by introducing reaction classes in which the rates of dissociation depend on whether or not two kinase molecules are present. Both toy models involve exactly the same 24 states and 101 reactions, but the rate contants for the dissociation reactions in which kinase-kinase interactions are present are reduced 100-fold in the second model. The 13 reaction classes defined in `toy.in` are expanded to 23 reaction classes in `toy_coop.in`. The cooperative interactions increase receptor aggregation about 20% at steady state, while kinase phosphorylation increases about 25-fold, such that about 25% of receptors are associated with an activated kinase (at steady state), as opposed to less than 1% in the absence of

[3]The TLRs each contain a conserved cytosolic protein sequence, the Toll/interleukin-1 receptor (TIR) domain, which plays a central role in signaling. The mechanism of signaling is similar for different TLRs as well as for other TIR-containing receptors, such as the interleukin-1 (IL-1) receptor (IL-1R). The TIR domain of a receptor interacts with a cytosolic adapter protein, such as MyD88. This adapter protein also interacts with a serine/threonine kinase, such as IL-1R associated kinase 1 (IRAK-1). Adapter-mediated coupling of IRAK-1 to a TIR-containing receptor mediates, through mechanisms yet to be fully characterized, hyperphosphorylation of IRAK-1, which is critical for downstream events. This simplified description of early events in signal transduction is elaborated in Fig. 1 if we associate the kinase, adapter, and receptor in this figure with IRAK-1, MyD88, and a TIR-containing receptor that dimerizes through receptor-receptor interaction in response to monovalent ligand-receptor binding. Thus, in the scheme of Fig. 1, which is highly speculative, the mechanism of IRAK-1 phosphorylation is ligand-induced dimerizaton of receptors that are each associated with MyD88 and IRAK-1. Co-localization of two molecules of IRAK-1 in this manner allows one to transphosphorylate the other. We caution that signaling is actually far more complicated. Activation of IRAK-1 is influenced by additional adapter proteins, such as Mal/TIRAP, and other members of the IRAK family, such as IRAK-4. For a recent review of signaling by a TLR, see [43].

cooperative interactions. The kinetics of kinase phosphorylation predicted by the two models are compared in panels (b) and (c) of Fig. 7. Thus, as was also found in an analysis of the FcϵRI model [7], a mechanism for selectively recruiting kinase molecules to receptor aggregates is necessary to generate substantial kinase activation. In the absence of such a mechanism, a substantial fraction of receptors (14% in the case of `toy.in`) may bind kinase, but only a minuscule fraction of these kinases are activated at steady state. Interestingly, ligand-induced recruitment of the adapter, which is consistent with observed recruitment of MyD88 during signaling by TIR-containing receptors [46], is predicted by the second model but not the first. This finding supports the idea that cooperative binding of signaling molecules may be a feature of early events in signaling by TIR-containing receptors. Analysis of the two toy models demonstrates that numerical values of rate constants can have a dramatic effect on the behavior of a signaling network, because the two reaction networks are otherwise identical. Thus, any method of analysis that does not consider quantitative aspects of molecular interactions may miss biophysical mechanisms that play a key role in signal processing.

8 MASS DEPENDENCE OF RATE LAWS

One question that arises about the approach taken in BioNetGen is whether it is correct to assume that the same rate constant applies to all reactions in a given class. It is possible, for example, that cooperative interactions among the molecules in a complex will affect binding rates. Unfortunately, it is often the case that no data is available about cooperativity. In the absence of data, the simplest assumption is that of zero cooperativity, i.e., no interaction between proteins that bind simultaneously to another protein. This assumption is the starting point for a search for cooperativity in that it serves as a null hypothesis.

The form of cooperativity perhaps of primary concern (and the form that would probably be easiest to predict or detect) is negative cooperativity arising from steric clashes. Although steric effects could prevent large proteins from binding to nearby binding sites on the same protein, multiple signaling proteins have been observed to bind non-competitively to a single scaffold-like pro-

tein [47]. Also, relatively large domains of a protein have been observed to interact simultaneously with much smaller closely-spaced binding sites. For example, the two tandem SH2 domains of the PTK Syk can simultaneously bind separate phosphotyrosines within a immunoreceptor tyrosine-based activation motif (ITAM) [48, 49]. Thus, steric constraints on complex formation may not be as severe as one might first think.

Computational studies of protein structure may have the potential to generate data that could be useful when developing reaction rules in the absence of other data. For example, homology modeling and molecular docking could be used to predict steric clashes [50, 51] and structural information could be used to predict the parameters of a reaction as a function of molecular context [52, 53].

Another factor that might cause variability in the rate of bimolecular reactions within a reaction class is the slower diffusion rate of larger complexes in comparison with either smaller complexes or individual molecules. Of course, differences in the rate of diffusion will only matter when reactions are affected by diffusion, which is not the case for binding in the reaction-limited regime (where the rate of molecular collision is much faster than the rate of chemical transformation). We expect reaction-limited binding for reactions between molecules in the cytosol or between an extacellular or cytosolic molecule and a membrane-anchored molecule when the concentration of membrane-anchored molecules is not too high [54]. The binding kinetics of an extracellular or cytosolic ligand associating or dissociating with a membrane protein will be influenced by diffusion of the ligand when the membrane protein concentrations are sufficiently high [55, 56, 57]. Likewise, enzyme-catalyzed reactions are often reaction-limited, although some enzymes catalyze reactions near the maximum, diffusion-limited rate [58]. For reactions between membrane-anchored molecules, there are some cases where diffusion effects can influence reactions, but passive changes in the diffusion coefficient of a transmembrane protein due to the addition of extracellular or cytosolic proteins are expected to be negligible, as explained below.

The viscosity of a cell membrane is at least an order of magnitude higher than the viscosity of the cytosol or extracellular medium (cf. [59, 60]), and thus, the attachment of molecules to the

extracellular or cytosolic region of a membrane protein are predicted to have a negligible effect on the rate of diffusion. Experimental measurements support this prediction [61, 62]. Changing the effective size of the transmembrane region, for example when a membrane protein associates with another membrane protein to form a dimer, is also likely to cause only a minor change in the diffusion coefficient because the diffusion coefficient has only a logarithmic dependence on membrane surface area [63]. This prediction is supported by experimental data indicating that monomers and dimers of FcϵRI have similar diffusion coefficients [64]. Interestingly, the diffusion coefficient decreases significantly for trimers of FcϵRI, which has been attributed to strong interactions of receptor trimers with the cytoskeleton as a result of receptor signaling [65]. Such an effect should certainly be included in any realistic model of a system, but this effect is not passive and cannot be predicted from diffusion theory.

Let us now consider the case of a diffusion-limited reaction in the solution phase. The Smoluchowski diffusion-limited forward rate constant, k_+, for the binding of two globular proteins in solution [66, 67] is given by

$$k_+ = 4\pi(D_1 + D_2)(a_1 + a_2), \tag{2}$$

where D_1 and D_2 are the diffusion coefficients of the two proteins and a_1 and a_2 are the contact radii of the proteins. Using the Stokes-Einstein equation, $D = k_B T/(6\pi\eta a)$, which relates the diffusion coefficient D of a protein to the Boltzmann constant k_B, the absolute temperature T, the medium viscosity η, and the protein radius a, and the geometric result that for a globular (i.e., spherical) protein $a \propto m^{1/3}$, where m is the mass of the protein, we find

$$k_+ = \frac{2k_B T}{3\eta}\left(2 + \left(\frac{m_1}{m_2}\right)^{1/3} + \left(\frac{m_2}{m_1}\right)^{1/3}\right). \tag{3}$$

This equation gives the correction needed for diffusion-limited binding reactions between molecules that may be associated with other factors. Consider, for example, binding of two proteins of equal molecular weight. When one of these two proteins is bound to a third protein that is ten times its mass, then k_+ for the binding reaction is reduced. However, the reduction is only 17% relative to

the case in which the third protein is absent. Such corrections are minor in general because k_+ varies sublinearly with the mass of a protein, as indicated in Eq. 3.

Based on the above considerations, we conclude that no theoretical correction for the effect of complexation on diffusion is necessary for reactions that involve only membrane proteins. Such corrections could well be important, as we have noted, but these corrections cannot be obtained from diffusion theory alone. For reactions that involve proteins diffusing in three dimensions, corrections will be relatively small because they depend sublinearly on mass differences, i.e., the correction is proportional to the cube root of the mass difference. Moreover, these corrections are needed only when the rates of reactions are affected by diffusion, which is atypical. Therefore, in BioNetGen 1.1 rate constants are not adjusted for mass differences among reactants within a reaction class. This feature could be added if required.

9 GRAPH EXTENSION OF STRING-BASED REPRESENTATION

As mentioned earlier, a limitation of BioNetGen is the inability to explicitly and systematically specify the connectivity of molecular components in a complex. The representational scheme of Faeder et al. [19] was proposed to remove this limitation. In this scheme, molecules are represented as graph partitions, which are drawn as boxes containing components, as in Fig. 1. Graphs comprise nodes, node labels, and edges. The nodes represent components of molecules, the text labels of these nodes give the names of components and the states of components (if any are specified), and edges represent bonds between components. A list of possible allowed states is provided when a component is introduced. A complex is represented as a graph, as in the case of a molecule, but the graph has more than one partition, one for each molecule in the complex. Complexes could also be represented as in BioNetGen through the state description of components, although this usage is deprecated. Regular expressions in reaction rules are replaced with graphs, which are subgraphs of graphs that represent chemical species. A missing label for the state of a component

is equivalent to the wildcard $*$ in BioNetGen.

According to the graphical conventions of model specification [19], reaction rules take the form of graph rewriting rules, cut-and-paste operations on graphs that transform a set of left (reactant) graphs into a set of right (product) graphs. (Cut-and-paste operations can be equivalent to relabeling operations.) To establish correspondence between reactant and product graphs, one will need to specify a mapping of reactant graph nodes to product graph nodes and indicate which reactant graph elements are and are not affected by graph rewriting. Parts of a reactant graph that are not affected by rewriting are purely contextual, e.g., an enzyme may need to be present to catalyze a reaction but the enzyme would be unaffected by the reaction.

The details of a procedure for applying graphical reaction rules have yet to be worked out; however, two classical problems must be solved in this procedure. One must be able to determine if two graphs are identical, and one must be able to determine if a graph representing a group pattern is an isomorphic subgraph of a second graph representing a chemical species. The first problem (graph isomorphism) must be solved to build and maintain the lists of chemical species and reactions. The second problem (subgraph isomorphism) must be solved to find chemical species that qualify as reactants as defined in a reaction rule. A canonical labeling of graphs [68], analogous to the sort function employed here for species labeling, may be useful for solving the problems of graph isomorphism. The method of Ullmann [69] may be useful for solving the problems of subgraph isomorphism, especially because we are only interested in mappings between nodes with identical name labels and so we can reject many mappings otherwise possible. The method of Ullmann allows information about name labels to be used before beginning the search for mappings of a subgraph into a graph.

DISCUSSION

In this paper, we have discussed the rationale behind our rule-based approach to modeling a biochemical network that is marked by combinatorial complexity, and we have outlined the steps

involved in specifying a model and interpreting the specification of a model. The method presented here, which we have illustrated using simple examples, is akin to cellular automata and agent-based modeling approaches, such as that of STOCHSIM [15], in that rules, which describe local interactions of the components in a system (e.g., modular protein-protein interactions of a signal-transduction network), determine the emergent behavior of the system. These rules are used to generate a list of reactions and participating chemical species. Thus, the rules and the process of rule generation lead to a physicochemical model, which can provide the basis for simulation studies and analysis of system behavior.

This report also serves to announce new features of the most recent release of the BioNetGen software package, a general-purpose tool for rule-based modeling. The current version of BioNetGen (1.1) differs from the original version (1.0) [17] in that now chemical species need not be enumerated before the application of reaction rules, network simulation can proceed without exhaustive generation of the network, and complexes consisting of more than two molecules having multi-state descriptions may be included in a model, although representation of complexes can still be problematic.

Because current conventions of model specification do not allow the connectivity of molecules in a complex to be represented explicity or systematically, we have proposed new graph-based conventions for model specification [19], which have been discussed and illustrated here. We have pointed out the new problems of reaction rule processing that arise with these conventions. The capability of the graph-based scheme to represent connectivity comes at the expense of needing to solve problems of graph and subgraph isomorphism in the application of reaction rules rather than the simpler problems of string matching encountered in the current implementation of BioNetGen. In the graphical method of representation, the reaction rules take the form of graph rewriting rules, with subgraphs replacing the regular expressions of BioNetGen. This new approach to model specification is inspired by the use of graphs and graph rewriting rules to model chemical systems (for examples, see [20, 21, 70, 71, 72, 73]). The literature about the use of rules, including graph rewriting rules, to model systems is vast. However, the rule-based approach described here is

one of the first attempts to use rules to generate physicochemical models of systems of molecular biology, such as protein interaction networks.

The conventions of BioNetGen provide a language for describing biological systems that is concise (a BioNetGen input file and the actual list of reactions specified in the file are compared in [17]) and precise. A BioNetGen input file is precise in that the underlying reaction network of a signal-transduction system is unambiguously defined; it is also comprehensive in that the reactions considered in the model are all those implied by the specified types of protein interactions (i.e., by the set of specified reaction rules). Many other researchers are also concerned with how to represent biological systems, particularly signal-transduction systems [23]. A related concern is standardization of representational schemes for electronic exchange and archive purposes [74]. Kohn and co-workers [22, 75], for example, have developed a formalization of diagrammatic interaction maps, which are commonly used ad hoc to describe the interactions of a system of proteins or the effects of protein interactions. Others have introduced tools of computer science, such as process algebra for representing concurrent processes, in an attempt to develop a formal language of molecular biology [76, 77, 78]. Notably, graphs have been used to represent biological systems in the frameworks of rewriting logic [79, 80] and process algebra [81, 82]. Finally, in addition to BioNetGen, several software tools for computer-aided specification and generation of mathematical/computational models have been developed [14, 15, 16, 33, 41, 83, 84, 85, 86]. Such tools allow one to formulate and analyze models that are much larger than those that can be reasonably specified using only paper and pencil. BioNetGen, for example, has been used to formulate dynamical models of signal-transduction networks, like the FcϵRI model [6, 7], that account for protein phosphorylation states and protein complexes found in hundreds to thousands of chemical species [18]. We believe that this type of modeling capability, which is needed to overcome the problem of combinatorial complexity, must be further developed and will play an important role in understanding the behavior of a cell. Standards for representing and exchanging models that include multi-state molecules and multi-component complexes are currently being developed [87].

ACKNOWLEDGEMENTS

This work was supported by grants GM35556 and RR18754 from the National Institutes of Health and by the Department of Energy through contract W-7405-ENG-36. We thank an anonymous reviewer for helpful comments.

References

[1] Hunter, T. Signaling—2000 and beyond. Cell 2000, 100, 113–127.

[2] Bray, D. Molecular prodigality. Science 2003, 299, 1189–1190.

[3] Hlavacek, W.S.; Faeder, J.R.; Blinov, M.L.; Perelson, A.S.; Goldstein, B. The complexity of complexes in signal transduction. Biotechnol Bioeng 2003, 84, 783–794.

[4] Ptashne, M.; Gann, A. Genes & Signals; Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 2002.

[5] Pawson, T.; Nash, P. Assembly of cell regulatory systems through protein interaction domains. Science 2003, 300, 445–452.

[6] Goldstein, B.; Faeder, J.R.; Hlavacek, W.S.; Blinov, M.L.; Redondo, A.; Wofsy, C. Modeling the early signaling events mediated by FcϵRI. Mol Immunol 2002, 38, 1213–1219.

[7] Faeder, J.R.; Hlavacek, W.S.; Reischl, I.; Blinov, M.L.; Metzger, H.; Redondo, A.; Wofsy, C.; Goldstein, B. Investigation of early events in FcϵRI-mediated signaling using a detailed mathematical model. J Immunol 2003, 170, 3769–3781.

[8] Goldstein, B.; Faeder, J.R.; Hlavacek, W.S. Mathematical and computational models of immune-receptor signalling. Nat Rev Immunol 2004, 4, 445–456.

[9] Bray, D.; Lay, S. Computer-based analysis of the binding steps in protein complex formation. Proc Natl Acad Sci USA 1997, 94, 13493–13498.

[10] Wofsy, C.; Torigoe, C.; Kent, U.M.; Metzger, H.; Goldstein, B. Exploiting the difference between intrinsic and extrinsic kinases: implications for regulation of signaling by immunoreceptors. J Immunol 1997, 159, 5984–5992.

[11] Endy, D.; Brent, R. Modelling cellular behavior. Nature 2001, 409, 391–395.

[12] Arkin, A.P. Synthetic cell biology. Curr Opin Biotechnol 2001, 12, 638–644.

[13] Hatzimanikatis, V.; Li, C.; Ionita, J.A.; Broadbelt, L.J. Metabolic networks: enzyme function and metabolite structure. Curr Opinion Struct Biol 2004, 14, 300–306.

[14] Morton-Firth, C.J.; Bray, D. Predicting temporal fluctuations in an intracellular signalling pathway. J Theor Biol 1998, 192, 117–128.

[15] Le Novère, N.; Shimizu, T.S. STOCHSIM: modelling of stochastic biomolecular processes. Bioinformatics 2001, 17, 575–576.

[16] Shimizu, T.; Aksenov, S.V.; Bray, D. A spatially extended stochastic model of the bacterial chemotaxis signalling pathway. J Mol Biol 2003, 329, 291–309.

[17] Blinov, M.L.; Faeder, J.R.; Goldstein, B.; Hlavacek, W.S. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. Bioinformatics 2004, 20, 3289–3291.

[18] `http://cellsignaling.lanl.gov`

[19] Faeder, J.R.; Blinov, M.L.; Hlavacek, W.S. Graphical rule-based representation of signal-transduction networks. Accepted for publication in Proceedings of the 2005 ACM Symposium on Applied Computing. March 14–17, Santa Fe, NM.

[20] Benkö, G.; Flamm, C.; Stadler, P.F. A graph-based toy model of chemistry. J Chem Inf Comput Sci 2003, 43, 1085–1093.

[21] Klavins, E.; Ghrist, R.; Lipsky, D. Graph grammars for self assembling robotic systems. Proceedings of the 2004 IEEE International Conference on Robotics and Automation, 2004, 5, 5293–5300.

[22] Aladjem, M.I.; Pasa, S.; Parodi, S.; Weinstein, J.N.; Pommier, Y.; Kohn, K.W. Molecular interaction maps—a diagrammatic graphical language for bioregulatory networks. Sci STKE 2004, pe8.

[23] Franza, B.R. From play to laws: language in biology. Sci STKE 2004, pe9.

[24] Perelson, A.S. Some mathematical models of receptor clustering by multivalent ligands. In *Cell Surface Dynamics: Concepts and Models* Perelson, A.S.; DeLisi, C.; Wiegel, F.W. editors, marcel Dekker, New York, 1984, pp. 223–276.

[25] Barabási, A.-L.; Oltvai, Z.N. Network biology: understanding the cell's functional organization. Nat Rev Genet 2004, 5, 101–113.

[26] Clarke, B.L. Stability of complex reaction networks. *Advances in Chemical Physics*, 1980, vol 43. Wiley, New York.

[27] Papin, J.A.; Palsson, B.O. Topological analysis of mass-balanced signaling networks: a framework to obtain network properties including crosstalk. J Theor Biol 2004, 227, 283–297.

[28] Papin, J.A.; Palsson, B.O. The JAK-STAT signaling network in the human B cell: an extreme signaling pathway analysis. Biophys J 2004, 87, 37–46.

[29] Bortz, A.B.; Kalos, M.H.; Lebowitz, J.L. A new algorithm for Monte Carlo simulation of Ising spin systems. J Comput Phys 1975, 17, 10–18.

[30] Gillespie, D.T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J Comput Phys 1976, 22, 403–434.

[31] Gillespie, D.T. Exact stochastic simulation of coupled chemical reactions. J Phys Chem 1977, 81, 2340–2361.

[32] Faulon, J.-L.; Sault, A.G. Stochastic generator of chemical structure. 3. Reaction network generation. J Chem Inf Comput Sci 2001, 41, 894–908.

[33] `http://www.molsci.org/~lok/moleculizer/`

[34] Gibson, M.A.; Bruck, J. Efficient exact stochastic simulation of chemical systems with many species and many channels. J Phys Chem A 2000, 104, 1876–1889.

[35] Gillespie, D.T. Approximate accelerated stochastic simulation of chemically reacting systems. J Chem Phys 2001, 115, 1716–1733.

[36] Gillespie, D.T.; Petzold, L.R. Improved leap-size selection for accelerated stochastic simulation. J Chem Phys 2003, 119, 8229–8234.

[37] Rathinam, M.; Petzold, L.R.; Cao, Y.; Gillespie, D.T. Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. J Chem Phys 2003, 119, 12784–12794.

[38] Cao, Y.; Li, H.; Petzold, L. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. J Chem Phys 2004, 121, 4059–4067.

[39] Puchalka, J.; Kierzek, A.M. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of biochemical reaction networks. Biophys J 2004, 86, 1357–1372.

[40] Kholodenko, B.N.; Demin, O.V.; Moehren, G.; Hoek, J.B. Quantification of short term signaling by the epidermal growth factor receptor. J Biol Chem 1999, 274, 30169–30181.

[41] Shapiro, B.E.; Levchenko, A.; Meyerowtiz, E.M.; Wold, B.J.; Mjolsness, E.D. Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. Bioinformatics 2003, 19, 677–678.

[42] Li, Q.; Dinner, A.R.; Qi, S.; Irvine, D.J.; Huppa, J.B.; Davis, M.M.; Chakraborty, A.K. CD4 enhances T cell sensitivity to antigen by coordinating Lck accumulation at the immunological synapse. Nat Immunol 2004, 5, 791–799.

[43] Pålsson-McDermott, E.M.; O'Neill, L.A.J. Signal transduction by the lipopolysaccharide receptor, Toll-like receptor-4. Immunology 2004, 113, 153–162.

[44] Akira, S.; Takeda, K.; Kaisho, T. Toll-like receptors: critical proteins linking innate and acquired immunity. Nat Immunol 2001, 2, 675–680.

[45] Takeda, K.; Kaisho, T.; Akira, S. Toll-like receptors. Annu Rev Immunol 2003, 21, 335–376.

[46] Bartfai, T.; Behrens, M.M.; Gaidarova, S.; Pemberton, J.; Shivanyuk, A.; Rebek, J., Jr. A low molecular weight mimic of the Toll/IL-1 receptor/resistance domain inhibits IL-1 receptor-mediated responses. Proc Natl Acad Sci USA 2003, 100, 7971–7976.

[47] Brooks, S.R.; Kirkham, P. M.; Freeberg, L.; Carter, R. H. Binding of cytoplasmic proteins to the CD19 intracellular domain is high affinity, competitive, and multimeric. J Immunol 2004, 172, 7556–7564.

[48] Ottinger, E.A.; Botfield, M.C.; Shoelson, S.E. Tandem SH2 domains confer high specificity in tyrosine kinase signaling. J Biol Chem 1998, 273, 729–735.

[49] Sada, K.; Takano, T.; Yanagi, S.; Yamamura, H. Structure and function of the Syk protein-tyrosine kinase. J Biochem 2001, 130, 177–186.

[50] Smith, G.R.; Sternberg, M.J.E. Prediction of protein-protein interactions by docking methods. Curr Opinion Struct Biol 2002, 12, 28–35.

[51] Tovchigrechko, A.; Wells, C.A.; Vakser, I.A. Docking of protein models. Protein Sci 2002, 11, 1888–1896.

[52] Gabdoulline, R.R.; Wade, R.C. Biomolecular diffusional association. Curr Opinion Struct Biol 2002, 12, 204–213.

[53] Schlosshauer, M.; Baker, D. Realistic protein-protein association rates from a simple diffusional model neglecting long-range interactions, free energy barriers, and landscape ruggedness. Protein Sci 2004, 13, 1660–1669.

[54] Lauffenburger, D.A.; Linderman, J.J. *Receptors: Models for Binding, Trafficking, and Signaling* Oxford University Press, New York, 1993, p. 146.

[55] Berg, H.C.; Purcell, E.M. Physics of chemoreception. Biophys J 1977, 20, 193–219.

[56] Erickson, J.; Goldstein, B.; Holowka, D.; Baird, B. The effect of receptor density on the forward rate constant for binding of ligands to cell surface receptors. Biophys J 1987, 52, 657–662.

[57] Haugh, J.M.; Lauffenburger, D.A. Physical modulation of intracellular signaling processes by locational regulation. Biophys J 1997, 72, 2014–2031.

[58] Ferscht, A. *Enzyme Structure and Mechanism*, 1985, 3rd edition, W.H. Freeman & Company, New York.

[59] Brown, E.D.; Wu, E.S.; Zipfel, W.; Webb, W.W. Measurement of molecular diffusion in solution by multiphoton fluorescence photobleaching recovery. Biophys J 1999, 77, 2837–2849.

[60] Pralle, A.; Keller, P.; Florin, E.-L.; Simons, K.; Hörber, J.K.H. Sphingolipid–cholesterol rafts diffuse as small entities in the plasma membrane of mammalian cells. J Cell Biol 2000, 148, 997–1007.

[61] Wolf, D.E.; Edidin, M.; Dragstein P.R. Effect of bleaching light on measurements of lateral diffusion in cell membranes by the fluorescence photobleaching recovery method. Proc Natl Acad Sci USA 1980, 77, 2043–2045.

[62] McCloskey, M.A.; Liu, Z.-Y.; Poo M.-M. Lateral electromigration and diffusion of Fcϵ receptors on rat basophilic leukemia cells: effects of IgE binding. J Cell Biol 1984, 99, 778–787.

[63] Saffman, P.G.; Delbrück, M. Brownian motion in biological membranes. Proc Natl Acad Sci USA 1975, 72, 3111–3113.

[64] Menon, A.K.; Holowka, D.; Webb, W.W.; Baird, B. Clustering, mobility, and triggering activity of small oligomers of immunoglobulin E on rat basophilic leukemia cells. J Cell Biol 1986, 102, 534–540.

[65] Menon, A.K.; Holowka, D.; Webb, W.W.; Baird, B. Cross-linking of receptor-bound IgE to aggregatees larger than dimers leads to rapid immobilization. J Cell Biol 1986, 102, 541–550.

[66] von Smoluchowski, M. Versuch einer mathematischen Theorie der Koagulationskinetic kolloider Lösungen. Z Phys Chem 1917, 92, 129–168.

[67] Keizer, J. Nonequilibrium statistical thermodynamics and the effect of diffusion on chemical reaction rates. J Phys Chem 1982, 86, 5052–5067.

[68] McKay, B.D. Practical graph isomorphism. Congressus Numerantium 1981, 30, 45–87. Available at `http://cs.anu.edu.au/~bdm/nauty/PGI/`

[69] Ullmann, J.R. An algorithm for subgraph isomorphism. J. ACM 1976, 23, 31–42.

[70] Faulon, J.-L. Isomorphism, automorphism partitioning, and canonical labeling can be solved in polynomial-time for molecular graphs. J Chem Inf Comput Sci 1998, 38, 432–444.

[71] Klein, M.T.; Hou, G.; Quann, R.J.; Wei, W.; Liao, K.H.; Yang, R.S.H.; Campain, J.A.; Mazurek, M.A.; Broadbelt, L.J. BioMOL: a computer-assisted biological modeling tool for complex chemical mixtures and biological processes at the molecular level. Environ Health Perspect 2002, 110, supplement 6, 1025–1029.

[72] Benkö, G.; Flamm, C.; Stadler, P.F. Generic properties of chemical networks: artificial chemistry based on graph rewriting. Lect Note Artif Intell 2003, 2801, 10–19.

[73] Kniemeyer, O.; Buck-Sorlin, G.H.; Kurth, W. A graph grammar approach to artificial life. Artif Life 2004, 10, 413–431.

[74] Hucka, M.; Finney, A.; Sauro, H.M.; Bolouri, H.; Doyle, J.C.; Kitano, H.; and the rest of the SBML forum. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 2003, 19, 524–531.

[75] Kohn, K.W. Molecular interaction maps as information organizers and simulation guides. Chaos 2001, 11, 84–97.

[76] Regev, A.; Silverman, W.; Shapiro, E. Representation and simulation of biochemical processes using the π-calculus process algebra. In Altman, R.B.; Dunker, A.K.; Hunter, L.; Klein, T.E. editors, *Pacific Symposium on Biocomputing*, 2001, vol. 6, pp. 459–470, Singapore, World Scientific Press.

[77] Priami, C.; Regev, A.; Shapiro, E.; Silverman, W. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Inf Process Lett 2001, 80, 25–31.

[78] Roux-Rouquié, M.; Caritey, N.; Gaubert, L.; Rosenthal-Sabroux, C. Using the unified modelling language (UML) to guide the systemic description of biological processes and systems. BioSystems 2004, 75, 3–14.

[79] Eker, S.; Knapp, M.; Laderoute, K.; Lincoln, P.; Meseguer, J.; Sonmez, K. Pathway logic: symbolic analysis of biological signaling. Pac Symp Biocomput 2002, 400–412.

[80] Talcott, C.; Eker, S.; Knapp, M.; Lincoln, P.; Laderoute, K. Pathway logic modeling of protein functional domains in signal transduction. Pac Symp Biocomput 2004, 568–580.

[81] Danos, V.; Laneve, C. Core formal molecular biology. Lect Note Comput Sci 2003, 2618, 302–318.

[82] Danos, V.; Laneve, C. Graphs for core molecular biology. Lect Note Comput Sci 2003, 2602, 34–46.

[83] `http://www.csi.washington.edu/teams/modeling/projects/BALSA/`

[84] `http://www.csi.washington.edu/teams/modeling/projects/sigtran/`

[85] `http://contraintes.inria.fr/BIOCHAM/`

[86] Chabrier-Rivier, N.; Chiaverini, M.; Danos, V.; Fages, F.; Schächter, V. Modeling and querying biomolecular interaction networks. Theor Comput Sci 2004, 325, 25–44.

[87] `http://sbml.org` (See the Wiki about SBML Level 3 efforts, particularly the proposals of A. Finney and others related to multi-state and multi-component species.)

FOOTNOTES

[1]The physiological function of a signal-transduction network is to convert an environmental signal, such as the presence of an extracellular ligand of a cell-surface receptor, into cellular responses, such as changes in gene expression, which allow a cell to adapt to the demands of its environment.

[2]Component states can be introduced to represent different conformations or modified forms of a molecular component. For example, the enzymatic activity of a kinase domain might be up-regulated by phosphorylation of its activation loop, which causes a conformational change. To distinguish the inactive and active forms of such a kinase, we need to track its conformational state, or equivalently the phosphorylation state of its activation loop. An alternative to introducing a phosphorylation state would be to represent a phosphate group as a distinct component. If this approach is followed, it is important to distinguish between covalent and non-covalent bonds when specifying a model.

[3]The TLRs each contain a conserved cytosolic protein sequence, the Toll/interleukin-1 receptor (TIR) domain, which plays a central role in signaling. The mechanism of signaling is similar for different TLRs as well as for other TIR-containing receptors, such as the interleukin-1 (IL-1) receptor (IL-1R). The TIR domain of a receptor interacts with a cytosolic adapter protein, such as MyD88. This adapter protein also interacts with a serine/threonine kinase, such as IL-1R associated kinase 1 (IRAK-1). Adapter-mediated coupling of IRAK-1 to a TIR-containing receptor mediates, through mechanisms yet to be fully characterized, hyperphosphorylation of IRAK-1, which is critical for downstream events. This simplified description of early events in signal transduction is elaborated in Fig. 1 if we associate the kinase, adapter, and receptor in this figure with IRAK-1, MyD88, and a TIR-containing receptor that dimerizes through receptor-receptor interaction in response to monovalent ligand-receptor binding. Thus, in the scheme of Fig. 1, which is highly speculative, the mechanism of IRAK-1 phosphorylation is ligand-induced dimerizaton of receptors that are each associated with MyD88 and IRAK-1. Co-localization of two molecules of

IRAK-1 in this manner allows one to transphosphorylate the other. We caution that signaling is actually far more complicated. Activation of IRAK-1 is influenced by additional adapter proteins, such as Mal/TIRAP, and other members of the IRAK family, such as IRAK-4. For a recent review of signaling by a TLR, see [43].

TABLE 1

Summary of Network Properties

Model	File Name	Molecules	Species	Uni Reactions
FcϵRI Model [6, 7]	`fceri_net.in`	4	354	3680
Original EGFR Model w/o PLCγ [40]	`egfr_path.in`	5	18	37
Extended EGFR Model [3, 40]	`egfr_net.in`	5	356	3749
Toy Model (this paper, Fig. 1)	`toy.in`	4	24	101

These networks, and others, are available as BioNetGen input files at our web site [18]. The seminal model of Kholodenko et al. [40] describes EGF-stimulated activation of the guanine-nucleotide exchange factor Sos. The versions of this model cited in the table omit phospholipase Cγ (PLCγ), which was included in the original model but is not involved in Sos activation. The extended form of the model encompasses the chemical species identified in ref. [3] except that only dimers of EGFR in which each receptor is bound to a ligand are considered, as in the original model [40]. When all possible ligand-bound forms of a dimer are considered, there are 1232, instead of 356, chemical species that are possible [3].

TABLE 2

Parameter Values in Toy Model

Symbol	Value	Parameter
k_{+L}, k_{-L}	0.1	Forward and reverse rate constants for ligand-receptor binding
k_{+D}	1	Rate constant for ligand-induced receptor dimerization
k_{-D}	0.1	Rate constant for receptor dimer dissociation
k_{+A}, k_{-A}	0.1	Forward and reverse rate constants for receptor-adapter binding
k_{+K}, k_{-K}	0.1	Forward and reverse rate constants for adapter-kinase binding
k_{+SK}, k_{-SK}	0.1	Forward and reverse rate constants for adapter-kinase complex binding to receptor
p_K	1	Rate of transphosphorylation catalyzed by unphosphorylated kinase
p_{K_s}	10	Rate of transphosphorylation catalyzed by phosphorylated kinase
d_M	1	Rate of dephosphorylation at the membrane
d_C	10	Rate of dephosphorylation in the cytosol
L_T, R_T, A_T, K_T	1	Total concentrations of ligand, receptor, adapter, and kinase
L at $t = 0$	L_T	Initial concentration of free ligand

Parameter values are given in consistent units. The initial concentrations of membrane-associated and cytosolic species are the equilibrium values in the absence of ligand. $dL_T/dt = dR_T/dt = dA_T/dt = dK_T/dt = 0$ for all t, except a bolus of ligand is introduced at $t = 0$.

FIGURE CAPTIONS

FIGURE 1. A system represented using the graphical conventions for model specification [19]. This system consists of a monovalent extracellular ligand, a monovalent cell-surface receptor, a bivalent cytosolic adapter protein, and a cytosolic kinase. The receptor dimerizes through a receptor-receptor interaction that depends on ligand binding. The adapter binds the receptor and the kinase. When two kinases are juxtaposed through binding to receptor-associated adapter proteins, one of the kinases can transphosphorylate the second kinase. In this representational scheme, nodes of a graph represent components of a molecule. Each node is named. The label of a node gives the name and the state of the corresponding component (if the component is allowed to have multiple states). Edges that join nodes represent bonds between components; only bonds that can form or break during signaling are represented explicitly. Graphs are partitioned; each partition corresponds to a molecule. Partitions are indicated by boxes surrounding a collection of nodes. An empty node indicates a component that is unbound. A half-filled node represents a component that may be bound or unbound. A filled-node represents a component that is bound. The graphs in reaction rules are subgraphs.

FIGURE 2. Definitions of key terms used to describe the procedures of rule-based modeling.

FIGURE 3. Simple examples of reaction rule processing. These examples illustrate establishment of correspondence between reactants and products. In the first example, at top, two molecules in a complex are mapped to the products that result from dissociation of the complex. Note that the state of component 1 of one molecule of A in the complex $A(1,*).A(1,*)$ changes from 1 to 0 upon dissociation of the complex. This transition corresponds to dissociation of a receptor dimer, which is formed through interaction of two receptors with a bivalent ligand. Dimer dissociation occurs when one of the two ligand-receptor bonds breaks. In the second example, at bottom, a particular multi-state molecule $A(*,0)$, which may (or may not) be associated with additional molecules in a complex (as indicated by the wildcard $*$ appended to $A(*,0)$), associates with a

single-state molecule B to form a complex. The multi-state molecule $A(*,0)$ on the reactant side of the reaction is mapped to the same molecule on the product side of the reaction. Any additional molecules associated with $A(*,0)$ on the reactant side of the reaction are also mapped to the same molecules on the product side of the reaction, i.e., the wildcard $*$ on the reactant side of the reaction maps to the wildcard $*$ on the product side of the reaction. The single-state molecule B is assigned a null correspondence, because it is anniliated in the reaction, i.e., the free form of this molecule, B, is lost. The bound form of B is represented as state 1 of component 2 of multi-state molecule A. Note that the free form of component 2 of A is represented as state 0 of this component, and binding of B to $A(*,0)*$ is represented as a change of state of component 2 of A from 0 to 1.

FIGURE 4. Declarations of single-state species, multi-state species, and groups of multi-state species and complexes. The text declarations found in the BioNetGen input file `toy.in` [18] are illustrated using the graphical conventions of [19] and the icons introduced in Fig. 1. These declarations introduce six single-state species (the ones shown explicitly in panel (a)) and multi-state species that contain a receptor comprised of three components (panels (b) and (c)). As indicated in panel (b), the first and second components of a receptor each have two possible states, and the third component has four possible states. However, not all combinations of these states are allowed (panel (c)). The two declarations of panel (c) limit the number of multi-state species. The first declaration permits eight multi-state species in the group $R(*,0,*)$, all of which contain one receptor. The second declaration permits four symmetric species and six (4 choose 2) asymmetric species in the group $R(1,1,*).R(1,1,*)$, all of which contain two receptors in a complex.

FIGURE 5. A set of reaction rules. These 13 rules, which are illustrated using the graphical conventions of Fig. 1, are declared in the BioNetGen input file `toy.in` [18]. Rules (3)–(5) illustrate how the text-based conventions for model specification can be more verbose than the graph-based conventions. These rules together are equivalent to the single graphical reaction rule that is shown above them, and we can consider them to define only a single class of reaction in which all re-

versible reactions are parameterized by the same forward and reverse rate constants. Likewise, we can consider Rules (6) and (7) to define a single reaction class, and we can consider Rules (8) and (9) to define a related but distinct reaction class. More than one reaction rule must be specified to define each of these classes because of the way that cytosolic species are treated in the model specification: each cytosolic species is represented using a single-state declaration. In contrast, Rules (10) and (11) and Rules (12) and (13) show that distinct reaction classes can be declared for the same type of chemical transformation to account for an effect of molecular context on the rate of chemical transformation. Rules (10) and (11) indicate that the rate of transphosphorylation catalyzed by a kinase in a receptor complex is affected by the phosphorylation state of the kinase. It is upregulated if $p_{K_s} > p_K$. Rules (12) and (13) indicate that dephosphorylation of a kinase is affected by its location in the cell: the rate constant d_M applies when the kinase is localized at the membrane in a receptor complex, whereas the rate constant d_C applies when the kinase is in the cytsol. This distinction is relevant if the phosphatases that mediate dephosphorylation, which are considered only implicity in this model specification, are localized (e.g., anchored to the inner membrane or free to diffuse in the cytosol).

FIGURE 6. Examples of function evaluation rules. These rules are declared in the BioNetGen input file `toy.in` [18]. Each rule specifies a readout that is a sum of variables (concentrations) in the model. The `RecDim` readout corresponds to the number of receptor dimers. The `Rec-A` readout corresponds to the number of adapters bound to a receptor. The `Rec-K` readout corresponds to the number of kinases in a complex with a receptor. The `Rec-Kp` readout corresponds to the number of phosphorylated kinases in a complex with a receptor.

FIGURE 7. Readouts defined in Fig. 6 as a function of time. Calculations for panels (a) and (b) were performed using the BioNetGen software package and are based on the parameter values of Table 2 and the model specification (`toy.in` [18]) illustrated in Figs. 4 and 5. Calculations for panel (c) were also performed using BioNetGen and are based on the alternative model with coop-

erativity added as discussed in the text (`toy_coop.in` [18]). Panel (a) shows all four readouts of Fig. 6 on the same scale. Panel (b) shows only the `Rec-Kp` readout; the scale has been magnified. Panel (c) shows how the readouts of panel (a) change when the model is modified.

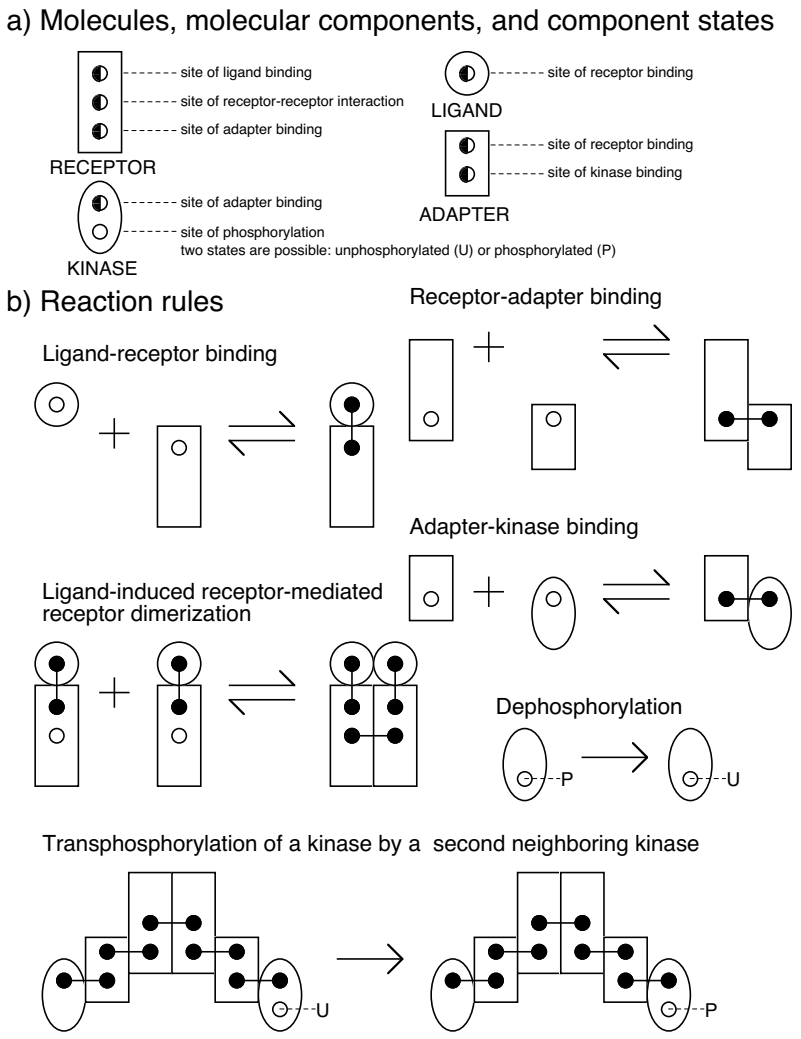

Figure 1: A system represented using the graphical conventions for model specification [19]. This system consists of a monovalent extracellular ligand, a monovalent cell-surface receptor, a bivalent cytosolic adapter protein, and a cytosolic kinase. The receptor dimerizes through a receptor-receptor interaction that depends on ligand binding. The adapter binds the receptor and the kinase. When two kinases are juxtaposed through binding to receptor-associated adapter proteins, one of the kinases can transphosphorylate the second kinase. In this representational scheme, nodes of a graph represent components of a molecule. Each node is named. The label of a node gives the name and the state of the corresponding component (if the component is allowed to have multiple states). Edges that join nodes represent bonds between components; only bonds that can form or break during signaling are represented explicitly. Graphs are partitioned; each partition corresponds to a molecule. Partitions are indicated by boxes surrounding a collection of nodes. An empty node indicates a component that is unbound. A half-filled node represents a component that may be bound or unbound. A filled-node represents a component that is bound. The graphs in reaction rules are subgraphs.

Component A part of a biomolecule, such as a site of post-translational modification, a motif, or a conserved recognition or catalytic domain in a protein.

State An attribute of a component, such as its state of non-covalent ligation, conformation, or covalent modification.

Bond A connection between components.

Molecule A set of components that form a unit, such as the components of a single polypeptide chain or a multimeric protein.

Molecule String Text consisting of a name for a molecule and an ordered list of indices that indicate the states of components of the molecule.

Chemical Species A molecule with each of its components in a particular state or a complex of molecules, each with components in particular states and connected through a particular configuration of bonds.

Group Pattern A pattern that identifies shared component states and bonds of a set of chemical species.

Reaction Rule A definition of a class of reaction, which may be associated with a particular type of interaction between components. A rule consists of a rate law and group patterns that can be used to identify sets of reactants and sets of products that result from reactions.

Function Evaluation Rule A mapping of the properties of a given set of chemical species, identified by one or more group patterns, into the variables of a given mathematical function, such as the sum of the concentrations of all species containing a particular molecule.

Figure 2: Definitions of key terms used to describe the procedures of rule-based modeling.

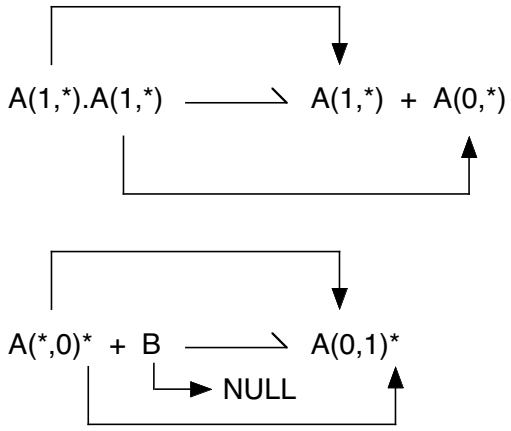

Figure 3: Simple examples of reaction rule processing. These examples illustrate establishment of correspondence between reactants and products. In the first example, at top, two molecules in a complex are mapped to the products that result from dissociation of the complex. Note that the state of component 1 of one molecule of A in the complex $A(1,*).A(1,*)$ changes from 1 to 0 upon dissociation of the complex. This transition corresponds to dissociation of a receptor dimer, which is formed through interaction of two receptors with a bivalent ligand. Dimer dissociation occurs when one of the two ligand-receptor bonds breaks. In the second example, at bottom, a particular multi-state molecule $A(*,0)$, which may (or may not) be associated with additional molecules in a complex (as indicated by the wildcard $*$ appended to $A(*,0)$), associates with a single-state molecule B to form a complex. The multi-state molecule $A(*,0)$ on the reactant side of the reaction is mapped to the same molecule on the product side of the reaction. Any additional molecules associated with $A(*,0)$ on the reactant side of the reaction are also mapped to the same molecules on the product side of the reaction, i.e., the wildcard $*$ on the reactant side of the reaction maps to the wildcard $*$ on the product side of the reaction. The single-state molecule B is assigned a null correspondence, because it is anniliated in the reaction, i.e., the free form of this molecule, B, is lost. The bound form of B is represented as state 1 of component 2 of multi-state molecule A. Note that the free form of component 2 of A is represented as state 0 of this component, and binding of B to $A(*,0)*$ is represented as a change of state of component 2 of A from 0 to 1.

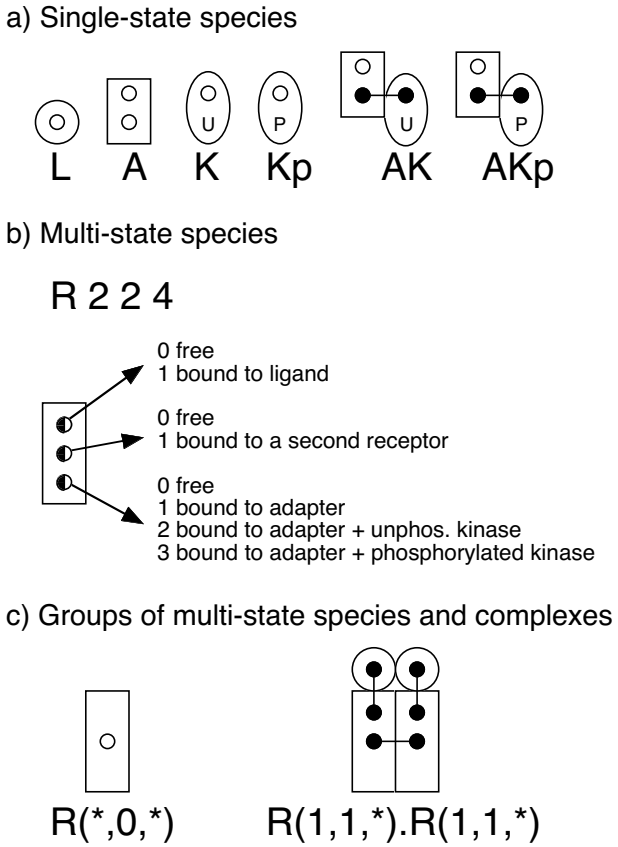

Figure 4: Declarations of single-state species, multi-state species, and groups of multi-state species and complexes. The text declarations found in the BioNetGen input file `toy.in` [18] are illustrated using the graphical conventions of [19] and the icons introduced in Fig. 1. These declarations introduce six single-state species (the ones shown explicitly in panel (a)) and multi-state species that contain a receptor comprised of three components (panels (b) and (c)). As indicated in panel (b), the first and second components of a receptor each have two possible states, and the third component has four possible states. However, not all combinations of these states are allowed (panel (c)). The two declarations of panel (c) limit the number of multi-state species. The first declaration permits eight multi-state species in the group $R(*,0,*)$, all of which contain one receptor. The second declaration permits four symmetric species and six (4 choose 2) asymmetric species in the group $R(1,1,*).R(1,1,*)$, all of which contain two receptors in a complex.

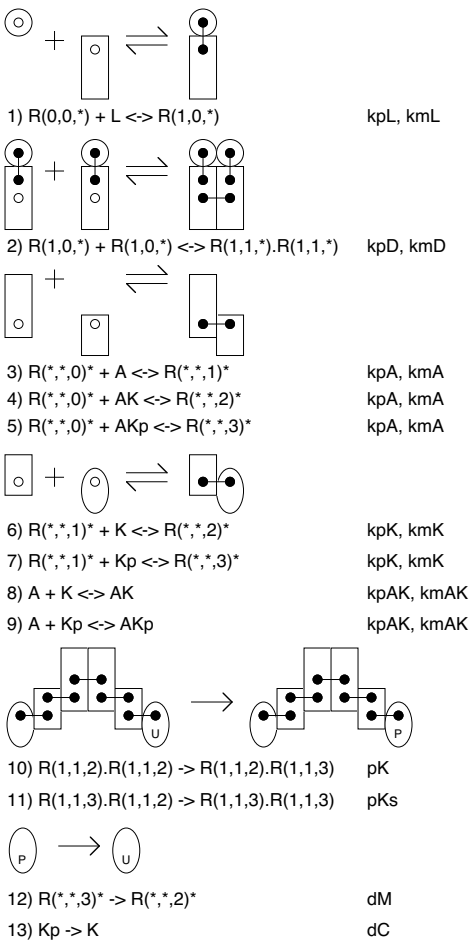

Figure 5: A set of reaction rules. These 13 rules, which are illustrated using the graphical conventions of Fig. 1, are declared in the BioNetGen input file `toy.in` [18]. Rules (3)–(5) illustrate how the text-based conventions for model specification can be more verbose than the graph-based conventions. These rules together are equivalent to the single graphical reaction rule that is shown above them, and we can consider them to define only a single class of reaction in which all reversible reactions are parameterized by the same forward and reverse rate constants. Likewise, we can consider Rules (6) and (7) to define a single reaction class, and we can consider Rules (8) and (9) to define a related but distinct reaction class. More than one reaction rule must be specified to define each of these classes because of the way that cytosolic species are treated in the model specification: each cytosolic species is represented using a single-state declaration. In contrast, Rules (10) and (11) and Rules (12) and (13) show that distinct reaction classes can be declared for the same type of chemical transformation to account for an effect of molecular context on the rate of chemical transformation. Rules (10) and (11) indicate that the rate of transphosphorylation catalyzed by a kinase in a receptor complex is affected by the phosphorylation state of the kinase. It is upregulated if $p_{K_s} > p_K$. Rules (12) and (13) indicate that dephosphorylation of a kinase is affected by its location in the cell: the rate constant d_M applies when the kinase is localized at the membrane in a receptor complex, whereas the rate constant d_C applies when the kinase is in the cytsol. This distinction is relevant if the phosphatases that mediate dephosphorylation, which are considered only implicity in this model specification, are localized (e.g., anchored to the inner membrane or free to diffuse in the cytosol).

```
Molecules    RecDim   R(*,1,*).*
Molecules    Rec-A    R(*,*,1)* R(*,*,2)* R(*,*,3)*
Molecules    Rec-K    R(*,*,2)* R(*,*,3)*
Molecules    Rec-Kp   R(*,*,3)*
```

Figure 6: Examples of function evaluation rules. These rules are declared in the BioNetGen input file `toy.in` [18]. Each rule specifies a readout that is a sum of variables (concentrations) in the model. The `RecDim` readout corresponds to the number of receptor dimers. The `Rec-A` readout corresponds to the number of adapters bound to a receptor. The `Rec-K` readout corresponds to the number of kinases in a complex with a receptor. The `Rec-Kp` readout corresponds to the number of phosphorylated kinases in a complex with a receptor.

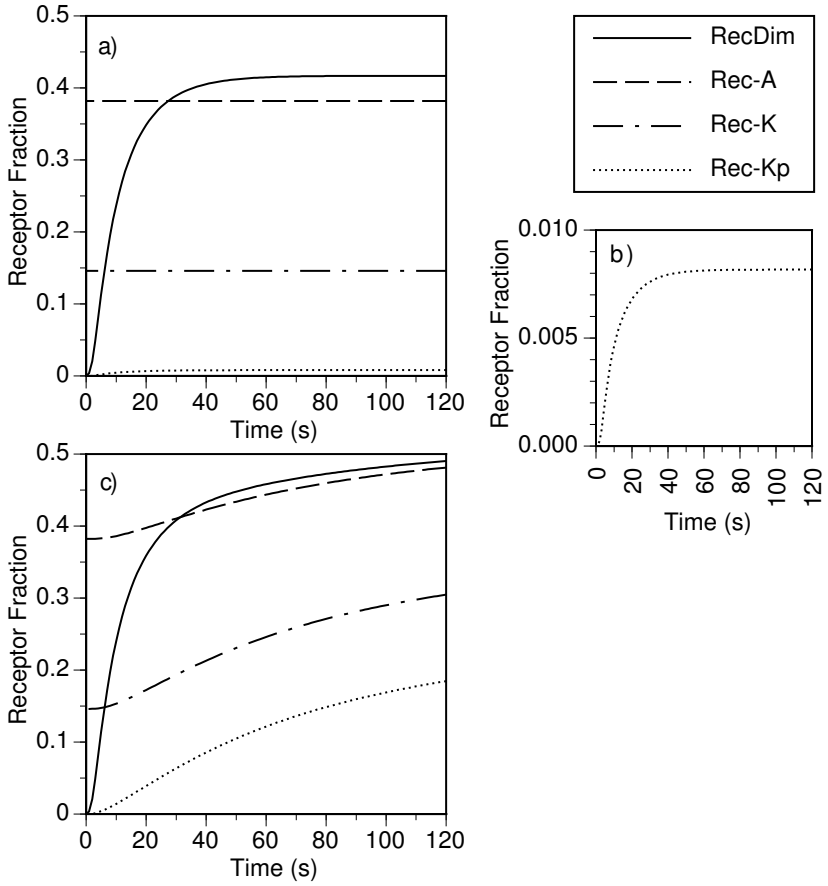

Figure 7: Readouts defined in Fig. 6 as a function of time. Calculations for panels (a) and (b) were performed using the BioNetGen software package and are based on the parameter values of Table 2 and the model specification (`toy.in` [18]) illustrated in Figs. 4 and 5. Calculations for panel (c) were also performed using BioNetGen and are based on the alternative model with cooperativity added as discussed in the text (`toy_coop.in` [18]). Panel (a) shows all four readouts of Fig. 6 on the same scale. Panel (b) shows only the `Rec-Kp` readout; the scale has been magnified. Panel (c) shows how the readouts of panel (a) change when the model is modified.